Learning Automata with Infinite State Cardinality

fwood@robots.ox.ac.uk

Joint work with Nicholas Bartlett, David Pfau, Jonathan Huggins, Jan Gasthaus,
Yee Whye Teh, Lancelot James, Cedric Archambeau and others

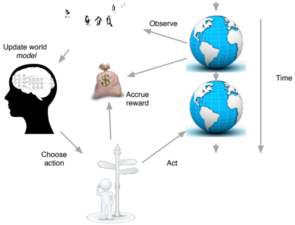
University of Oxford

June, 2013

Predictive Modelling

- Actors must have predictive models of worlds in order to plan
- Better predictive models can yield better plans and more rewarding sequences of actions

Ultimate goal: life-long learning algorithm that incrementally extracts a representation of the evolving world that is useful for planning from a single, continually growing sequence of observations (n.b. automata aren't this but are instructive)


Observe
Update world *model*
Accrue reward
Time
Choose action
Act

(Click to Play)

Computational Constraints Imposed By Life-Long Learning

- Incremental model estimation given a single observation sequence of growing (unbounded) length
- Estimator must have linear-time complexity
- The model must be representable in constant space
- Predictive inference must be performed in constant time

What is an automaton?

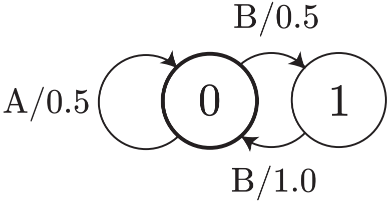
Classically

- An automaton is a finite representation of a formal language that may be an infinite set.
- Automata are often classified by the class of formal languages they are able to recognise.
- The set of all the words accepted by an automaton is called the language recognised by the automaton.

Today

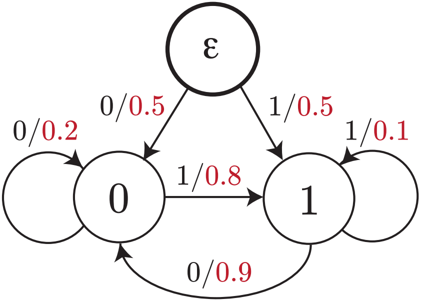
- Automata with unbounded state cardinality (little existing theory).
- Learning automata from data (given a production sequence, what was the automata that produced it?)

PDFA Notation $\neq$ Graphical Model



- What does this little probabilistic deterministic finite automata (PDFA) generate?
- What is a state?

A Familiar PDFA


ε
0/0.5
1/0.5
0/0.2
1/0.1
0
1/0.8
1
0/0.9

Automata corresponding to 1$^{\mathrm{st}}$-order Markov model over binary alphabet with fixed emission distributions

Notation

Formally a PDFA is a quintuple

$$M = (Q, \Sigma, \delta, \{G_{[i]}\}, q_0)$$

where

- Q is a finite set of states
- Σ is a finite set of symbols
- $\delta : Q \times \Sigma \to Q$ is a deterministic transition function from state/symbol pairs to another state
- $\{G_{[i]} : i \in Q\}$ is a set of probability vectors over symbols, one for each state
- q_0 is the initial state

Learning Problem(s)

Given a single, growing sequence $\mathbf{x} = x_1, x_2, \ldots, x_i, \ldots \in \Sigma^+$ of discrete observations one might like to learn

- $\{G_{[i]} : i \in Q\}$, the set of emission probability distributions with Q and δ fixed.

or

- $\{G_{[i]} : i \in Q\}$, the set of emission probability distributions and
- Q, the set of states and
- $\delta : Q \times \Sigma \rightarrow Q$, the transition function from state/symbol pairs to another state

Speculation

Automata

PDFA $\subset$ PNFA $\subset$ Probabilistic pushdown automata $\subset$ Turing machine

Corresponding Bayesian nonparametric models

SM [30], PDIA [21] $\subset$ IHMM [3, 8, 14] $\subset$ IPCFG [16, 18] $\subset$ Church [12]

- More smoothing in emissions distributions on the right, more complex latent state on the right
- Lessons learned from simpler models might inform development of more practical complex models
- Simpler models might be used to "smooth" program inference

Challenge Problem

Incrementally Learn a Model to Predict What Comes Next

01001001, 01101110, 00100000, 01110100,
01101000, 01110010, 01100101, 01100101,
00100000, 01100100, 01100001, 01111001,
01110011, 00100000, 01111001, 01101111,
01110101, 01110010, 00100000, 01101000,
01100001, 01110010, 01100100, 00100000,
01100100, 01110010, 01101001, 01110110,
01100101, 00100000, 01110111, 01101001,
01101100, 01101100, 00100000, 01100011,
01110010, 01100001, 01110011, 01101000 . . .

Graphical model for smoothing PDFA emission distributions

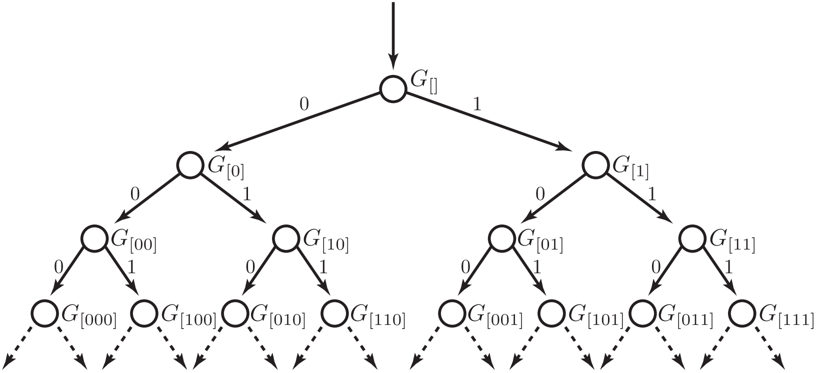

$G_{[]}$
0
1
$G_{[0]}$
$G_{[1]}$
0
1
0
1
$G_{[00]}$
$G_{[10]}$
$G_{[01]}$
$G_{[11]}$
0
1
0
1
0
1
0
1
$G_{[000]}$
$G_{[100]}$
$G_{[010]}$
$G_{[110]}$
$G_{[001]}$
$G_{[101]}$
$G_{[011]}$
$G_{[111]}$

Figure : *Graphical model* for coupling binary Markov model (= automata with fixed δ (not shown)) emission distributions

Binary Sequence Memoizer

Notation

$$\begin{aligned} G_{\varepsilon} | U_{\Sigma}, d_0 &\sim \mathrm{PY}(d_0, 0, U_{\Sigma}) \\ G_{\mathbf{u}} | G_{\sigma(\mathbf{u})}, d_{|\mathbf{u}|} &\sim \mathrm{PY}(d_{|\mathbf{u}|}, 0, G_{\sigma(\mathbf{u})}) \quad \forall \mathbf{u} \in \Sigma^{+} \\ x_t | x_1, \ldots, x_{t-1} = \mathbf{u} &\sim G_{\mathbf{u}} \end{aligned}$$

Here $\sigma(x_1 x_2 x_3 \ldots x_t) = x_2 x_3 \ldots x_t$ is the suffix operator, $\Sigma = \{0, 1\}$, $U_{\Sigma} = [.5, .5]$, and $\delta(x_1 x_2 \ldots x_{t-1}, x_t) \equiv x_1 x_2 \ldots x_{t-1} x_t$.

Encodes natural assumptions :

- Recency assumption encoded by form of coupling related conditional distributions ("back-off")
- Power-law effects encoded by the choice of hierarchical modeling glue: the Pitman-Yor process (PY).

Pitman-Yor process

A Pitman-Yor process $\mathrm{PY}(c, d, H)$ is a distribution over distributions with three parameters

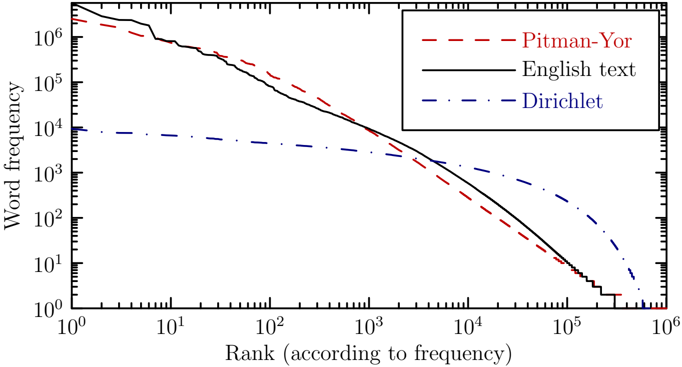
- A discount $0 \leq d < 1$ that controls power-law behavior
  - $d = 0$ is Dirichlet process (DP)
- A concentration $c > -d$ like that of the Dirichlet process (DP)
- A base distribution H also like that of the DP

Key properties :

If $G \sim \mathrm{PY}(c, d, H)$ then *a priori*

- $\mathbb{E}[G(s)] = H(s)$
- $\mathrm{Var}[G(s)] = (1 - d)H(s)(1 - H(s))$

Text and Pitman-Yor Process Power-Law Properties



Power-law scaling of word frequencies in English text. Relative word frequency is plotted against each words' rank when ordered according to frequency. There are a few very common words and a large number of relatively rare words.

Figure from [31].

Big Problem

$$G_{\mathbf{u}} | G_{\sigma(\mathbf{u})}, d_{|\mathbf{u}|} \sim \mathrm{PY}(d_{|\mathbf{u}|}, 0, G_{\sigma(\mathbf{u})}) \quad \forall \mathbf{u} \in \Sigma^{+}$$

Space. And time.

Coagulation and Fragmentation[1]

PY properties used to achieve linear space sequence memoizer.

Coagulation

If

$$G_2|G_1 \sim \mathrm{PY}(d_1, 0, G_1)$$

and

$$G_3|G_2 \sim \mathrm{PY}(d_2, 0, G_2)$$

then

$$G_3|G_1 \sim \mathrm{PY}(d_1 d_2, 0, G_1)$$

with G_2 marginalized out.

Fragmentation

Suppose

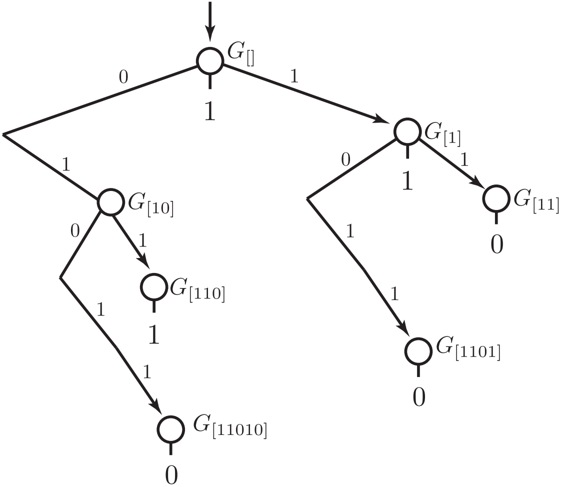
$$G_3|G_1 \sim \mathrm{PY}(d_1 d_2, 0, G_1).$$

Fragmentation operations exist to reinstantiate G_2 and G_3.

---

[1] [23, 13, 30]

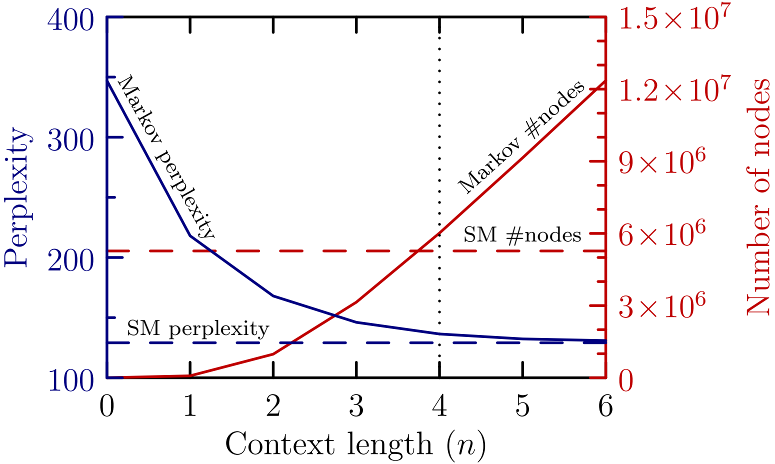
$O(T)$ Sequence Memoizer[2] Graphical Model for $\mathbf{x} = 110100$


$G_{[]}$
0
1
1
$G_{[1]}$
1
0
1
1
$G_{[10]}$
$G_{[11]}$
0
1
1
0
$G_{[110]}$
1
1
1
1
$G_{[1101]}$
1
0
$G_{[11010]}$
0

---

[2] [Wood et al, 2009]

$O(T)$ Sequence Memoizer vs. Smoothed n^{th}-order Markov Model



SM versus n^{th} order Markov models with hierarchical PYP priors as n varies. In red is computational complexity.

Problem

Nonparametric $\rightarrow O(T)$ space $\implies$ doesn't work for life-long learning

Towards Lifelong Learning : Incremental Estimation

Sequential Monte Carlo inference [Gasthaus, Wood, and Teh, 2010]

- Wikipedia 100MB → 20.8MB (1.66 bits/byte[3])
  - ≈ PAQ, special-purpose, non-incremental, Hutter-prize leader

Still $O(T)$ storage complexity.

---

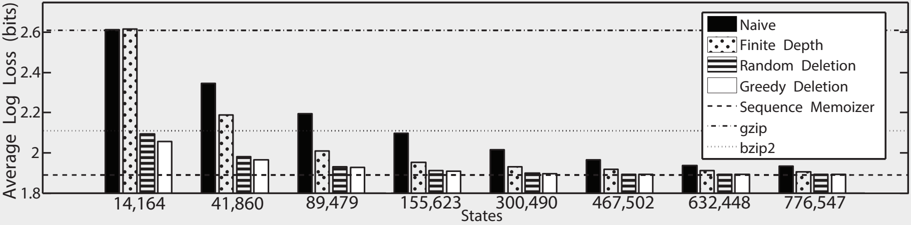
[3] average log-loss

$$-\frac{1}{T}\sum_{t=1}^{T}\log_2 P(x_t|\mathbf{x}_{1:t-1})$$

is average # of bits to transmit one byte under optimal entropy encoding (lower is better)

Towards Lifelong Learning : Constant Space Approximation

"Forgetting Counts" [Bartlett, Pfau, and Wood, 2010]

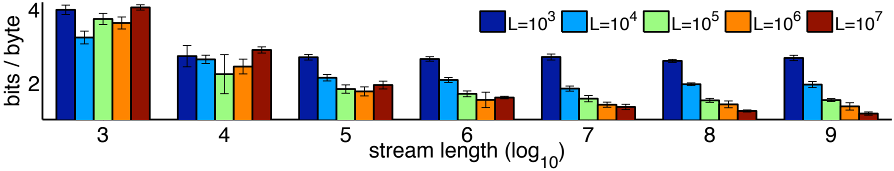


Constant-state-count SM approximation
Data: Calgary corpus, States = number of states used in 3-10 grams

"Improvements to the SM" [9]

Constant-size representation of predictive distributions

Life-Long Learning SM [Bartlett, Pfau, and Wood, 2010]



SM with life-long learning computational characteristics.

Wikipedia : 26.8GB → 4GB (≈ 1.2) bits/byte

L is number of states in SM, data Wikipedia dump 2010 head, depth 16,

How Good Is ≈ 1.5 bits/byte?

the fastest way to make money is ...

> *being player . bucketti wish i made it end in the reducers and assemblance smart practices to allow city is something to walk in most of the work of agriculture . i 'd be able to compete with an earlier goals : words the danger of conduct in southern california , has set of juice of the fights lack of audience that the eclasm beverly hills . " she companies or running back down that the book , " grass . and the coynes , exaggerate between 1972 . the pad , a debate on emissions on air political capital crashing that the new obviously program " – irock price , " coach began refugees , much and*[4]

---

[4] constant-space byte-SM, NY Times Corpus (50MB), log-loss 1.49 bits/byte, 2 million rest.'s

Summary So Far

- SM = powerful Bayesian nonparametric sequence model
  - Intuition : $\lim_{n \to \infty}$ smoothing *n*-gram
  - Byte predictive performance *in human range*
  - Bayesian-nonparametric $\implies$ incompatible with lifelong learning
  - Non-trivial approximations preserve performance
  - Resulting model compatible with lifelong learning
- Widely useful
  - Language-modeling
  - Lossless compression
  - Plug-in replacement for Markov model component of any complex graphical model
  - Direct plug-in replacement for *n*-gram language model
    - Software : `http://www.sequencememoizer.com/`

Software

A general purpose streaming lossless compressor ("deplump") built on the sequence memoizer is available for demonstration at

- http://www.deplump.com/
    - real time performance

Source code for the sequence memoizer is downloadable from

- http://www.sequencememoizer.com/

Can We Do Better?

Problem :

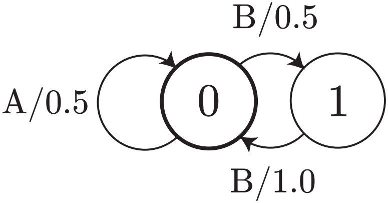
- No long-range coherence

Solution :

- Develop models that more efficiently encode long-range dependencies
    - Probabilistic deterministic infinite automata
    - Structured infinite hidden semi-Markov model
    - PCFG?
    - Probabilistic program?

PDFA Encode Long-Range Dependencies More Efficiently

Example probabilistic deterministic finite automata (PDFA) with no finite-order Markov model equivalent



- This **Even process** assigns positive probability to ABBA, AAABBBBABB, BBBBBBBB, but not to ABA

Probabilistic Deterministic Infinite Automata (PDIA)

A PDIA [Pfau, Bartlett, and Wood, 2011] is a Bayesian nonparametric PDFA with an unbounded number of states

Key Characteristics :

- Incrementally learnable
- Prior that biases automata complexity
- Structure learned from data[5]
- O(T) storage

Approximations for life-long learning :

- Future work

---

[5] Markov models are in the support of PDIA, can we exploit that?

Probabilistic Deterministic Infinite Automata (PDIA)

Formally a PDIA is a quintuple

$$M = (Q, \Sigma, \delta, \{G_i\}, q_0)$$

where

- Q is a countable set of states
- Σ is a finite set of symbols
- $\delta : Q \times \Sigma \to Q$ is a deterministic transition function from state/symbol pairs to another state
- $\{G_i : i \in Q\}$ is a set of probability vectors over symbols, one for each state
- q_0 is the initial state

A PDIA Prior

$$\begin{aligned} H_0 &\sim \mathsf{PY}(\gamma, d_0, H) \\ H_\sigma &\sim \mathsf{PY}(\alpha, d, H_0) \\ \delta(q_i, \sigma) &\sim H_\sigma \\ G_i &\sim \mathrm{Dir}(\beta/|\Sigma|, \ldots, \beta/|\Sigma|) \end{aligned}$$

given a sequence **x** a posterior over

$$\mathcal{H} = \{\delta, \{G_i\}\}$$

can be learned from data and used for prediction

$$P(x_T|\mathbf{x}) = \int P(x_T|\mathcal{H}) P(\mathcal{H}|\mathbf{x}) dP(\mathcal{H})$$

PDIA Inference

With state $\xi_t \in Q$ and $\xi_t = \delta(\xi_{t-1}, x_{t-1})$ the PDIA likelihood

$$P(\mathbf{x}|\{G_i\}, \delta) = G_{\xi_0}(x_0) \prod_{t=1}^{T} G_{\xi_t}(x_t)$$

- doesn't require marginalizing over paths (vs. HMM/PNFA) (fast)
- is of a form that $\{G_i\}$ can be marginalized out

Sampling δ requires "integrating" over paths diverging from the "current state" of the machine

PDIA Posterior Expectations

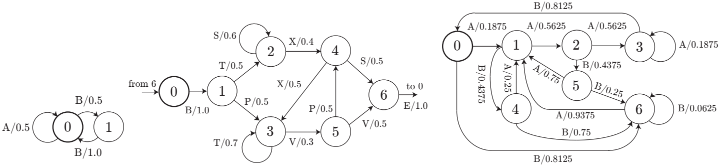
Posterior expectations "average over PDFA's".

- World states are *learned* from data.
- Prior biases towards a state re-use (other alternatives?)
- State-to-state transitions learned from data.
- States not identifiable by context (vs. Markov model)

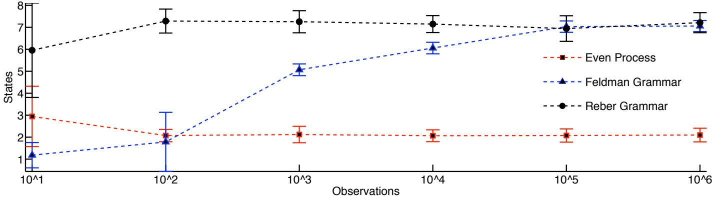
Lessons

- Smoothness: in order to "move around" space of PDIA's *a priori* every state must be able to reach every other state and every emission distribution must be able to emit everything.
- Smoothing: because relationships between states are unknown, hierarchical smoothing of emission distributions is not possible.

Learning PDFA


A/0.5
0
1
B/0.5
B/1.0
from 6
0
B/1.0
1
T/0.5
P/0.5
S/0.6
2
X/0.4
4
X/0.5
S/0.5
6
to 0
E/1.0
3
T/0.7
V/0.3
5
P/0.5
V/0.5
B/0.8125
0
A/0.1875
1
A/0.5625
2
A/0.5625
3
A/0.1875
B/0.4375
B/0.4375
A/0.25
A/0.75
5
B/0.25
4
A/0.9375
6
B/0.0625
B/0.75
B/0.8125

(a) Even (b) Reber (c) Feldman



Comparative Byte Prediction Performance

	PDIA	PDIA-MAP	HMM-EM	bigram	trigram	4-gram	5-gram	6-gram	SM (PF-1)
AIW	5.13	5.46	7.89	9.71	6.45	5.13	4.80	4.69	4.78
	365.6	379	52	28	382	2,023	5,592	10,838	19,358
DNA	3.72	3.72	3.76	3.77	3.75	3.74	3.73	3.72	3.56
	64.7	54	19	5	21	85	341	1,365	314,166

Table : PDIA inference performance (perplexity) relative to HMM and fixed order Markov models. Top rows: perplexity. Bottom rows: number of states in each model. For the PDIA this is an average number.

Results

- Expected : PDIA is more compact than the SM
- Unexpected : SM predicts best
- Expected : single HMM even more compact
- More Unexpected : single HMM predicts worst

Questions

- More data? Convergence rates? Wrong objective? Smoothing vs. compactness?

AIW - ≈10k char. train, ≈4k char. test; DNA ≈150k b.p. train, ≈50k b.p. test. Perplexity = $2^{\log-\text{loss}}$

PNFA $\neq$ PDFA

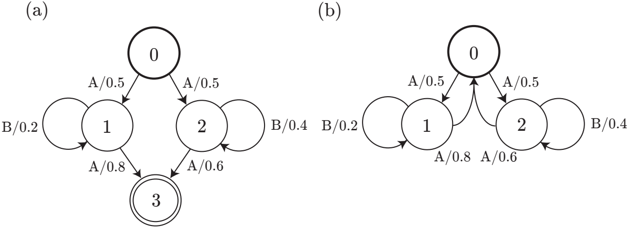

(a)
0
A/0.5
A/0.5
B/0.2
1
2
B/0.4
A/0.8
A/0.6
3
(b)
0
A/0.5
A/0.5
B/0.2
1
2
B/0.4
A/0.8
A/0.6

Figure : Two PNFAs outside the class of PDFAs. (a) can be represented by a mixture of two PDFAs, one following the right branch from state 0, the other following the left branch. (b), in contrast, cannot be represented by any finite mixture of PDFAs.

PNFA's can be transformed into HMM's and vice versa [6, 11].

Learning PNFA That Efficiently Encode Long Range Dependencies

Infinite Structured Hidden Semi-Markov M. [Huggins and Wood, to appear 2013.]

HMM with infinite number of states[6] + key additional characteristics :

- Explicitly parameterised duration distributions (like [15])
  - Allows for direct inference about long-range dependencies
- Prior transition constraints $\Longrightarrow$ a posteriori structured HMM
  - e.g. left-to-right infinite HMM

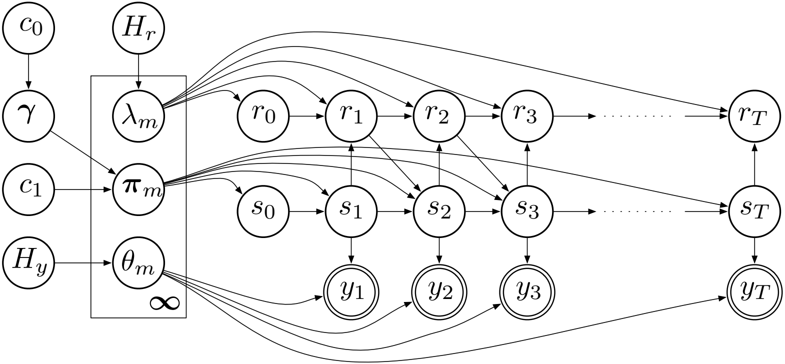
Approximations for life-long learning :

- Future work

---

[6] HDP-HMM [28] - HDP + SNΓP [24]

Infinite Structured Hidden Semi-Markov M.: Graphical Model


c_0
H_r
γ
λ_m
r_0
r_1
r_2
r_3
r_T
c_1
$\boldsymbol{\pi}_m$
s_0
s_1
s_2
s_3
s_T
H_y
θ_m
y_1
y_2
y_3
y_T
∞

Structured, dependent, infinite-dimensional transition distributions?

ISHSMM: Recipe

- Poisson process [17]
- Gamma process [17]
- SNΓPs [24]
- *Structured, dependent, infinite-dimensional* transition distributions [Huggins and Wood, to appear 2013.]

Infinite Structured Hidden Semi-Markov M.: Recipe

- Gamma process [17] as Poisson process over $\Theta \otimes V \otimes [0, \infty)$ with rate / mean measure

$$\mu(\tilde{\Theta}, \tilde{V}, \tilde{S}) = \alpha(\tilde{\Theta}, \tilde{V}) \int_{\tilde{S}} \gamma^{-1} e^{-\gamma}$$

- A draw from a Gamma process with

$$\alpha(\tilde{\Theta}, \tilde{V}) = c_0 H_\theta(\tilde{\Theta}) H_v(\tilde{V}).$$

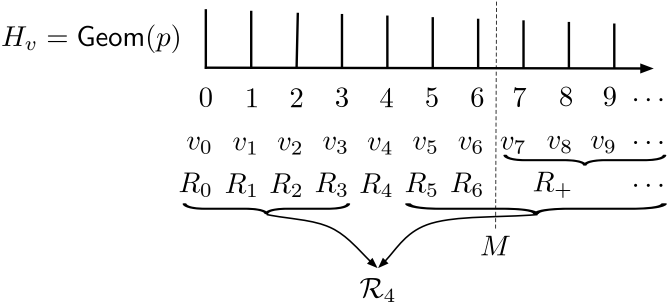
  [17] has the form

$$G = \sum_{m=1}^{\infty} \gamma_m \delta_{(\theta_m, v_m)}$$

  where $(\theta_m, v_m) \sim H_\theta \times H_v$.

Infinite Structured Hidden Semi-Markov M.: Recipe

- Non-disjoint "restricted projections" of Gamma processes are *dependent* Gamma processes (SNΓPs) [24]


$H_v = \mathsf{Geom}(p)$
0 1 2 3 4 5 6 7 8 9 ⋯
v_0 v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 ⋯
R_0 R_1 R_2 R_3 R_4 R_5 R_6 R_+ ⋯
M
$\mathcal{R}_4$

$$G_0 = \sum_{m \neq 0} \ldots, \quad \cdots \quad , G_4 = \sum_{m \neq 4} \gamma_m \delta_{\theta_m}, \quad \cdots$$

Infinite Structured Hidden Semi-Markov M.: Recipe

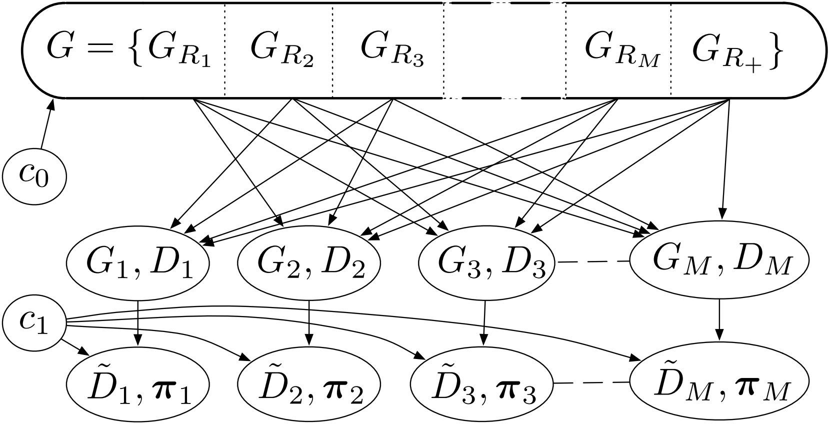
- Normalized dependent ΓP draws are dependent Dirichlet process draws.[7] In the ISEDHMM, DP draws are the dependent, structured, infinite-dimensional transition distributions

$$\begin{aligned} D_4 &= \frac{G_4}{G_4(\Theta)} \quad (1) \\ &= \frac{\sum_{m \neq 4} \gamma_m \delta_{\theta_m}}{\sum_{\theta \in \Theta} \sum_{m' \neq 4} \gamma_{m'} \delta_{\theta_{m'}}} \quad (2) \\ &= \frac{\sum_{m \neq 4} \gamma_m \delta_{\theta_m}}{\sum_{m' \neq 4} \gamma_{m'}} \quad (3) \\ &= \sum_{m \neq 4} \frac{\gamma_m}{\sum_{m' \neq 4} \gamma_{m'}} \delta_{\theta_m} \end{aligned}$$

---

[7] a draw from a Dirichlet process [7] is an infinite sum of weighted atoms [26] where the weights sum to one.

Infinite Structured Hidden Semi-Markov M. : Recipe


$G = \{G_{R_1}$
G_{R_2}
G_{R_3}
G_{R_M}
$G_{R_+}\}$
c_0
G_1, D_1
G_2, D_2
G_3, D_3
G_M, D_M
c_1
$\tilde{D}_1, \boldsymbol{\pi}_1$
$\tilde{D}_2, \boldsymbol{\pi}_2$
$\tilde{D}_3, \boldsymbol{\pi}_3$
$\tilde{D}_M, \boldsymbol{\pi}_M$

- Structured, dependent, infinite dimensional transition distributions $\boldsymbol{\pi}_m$ can be formed from draws from DDPs [Huggins and Wood, to appear 2013.]

Inference: Beam Sampling

We employ the forward-filtering, backward slice-sampling approach for the IHMM of [29] and EDHMM of [Dewar, Wiggins and Wood, 2012], in which the state and duration variables s and r are sampled conditioned on auxiliary slice variables u.

Net result: efficient, always finite forward backward procedure for sampling latent states.

Results

To illustrate SIHSMM learning on synthetic data, five hundred datapoints were generated using a 4 state EDHMM with Poisson duration distributions

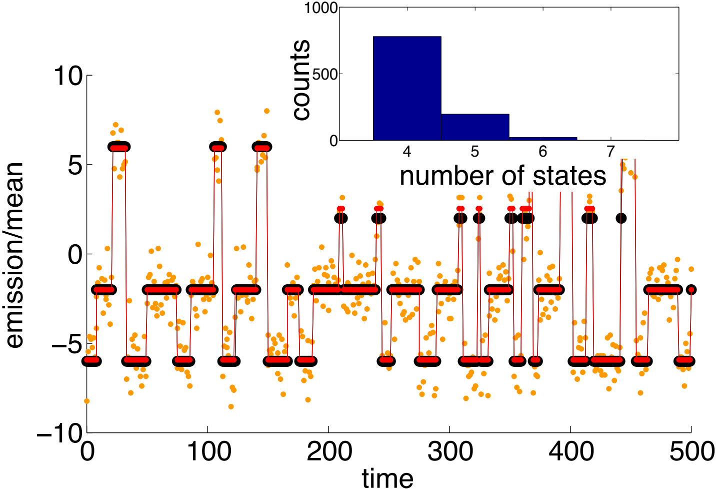
$$\boldsymbol{\lambda} = (10, 20, 3, 7)$$

and Gaussian emission distributions with means

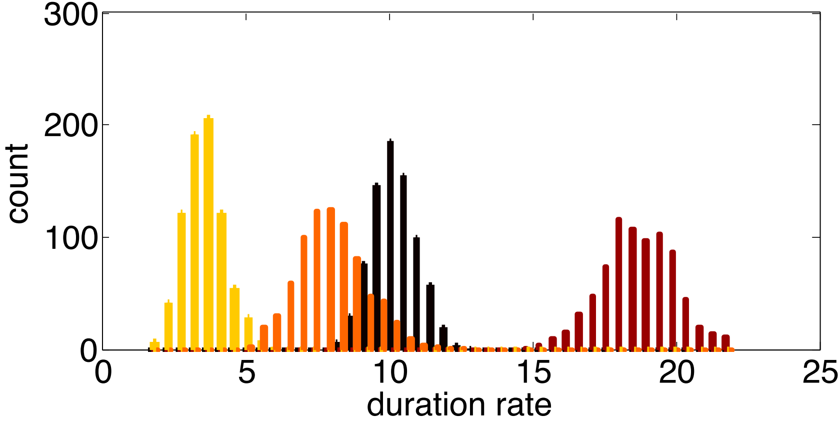
$$\boldsymbol{\mu} = (-6, -2, 2, 6)$$

all unit variance.

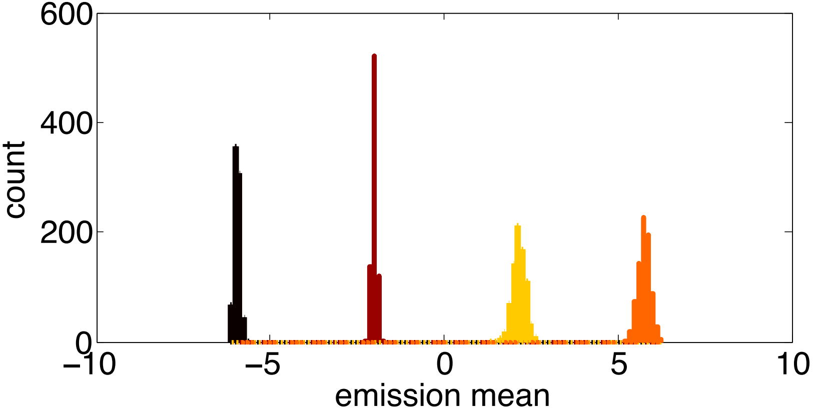
SIHSMM: Synthetic Data Results



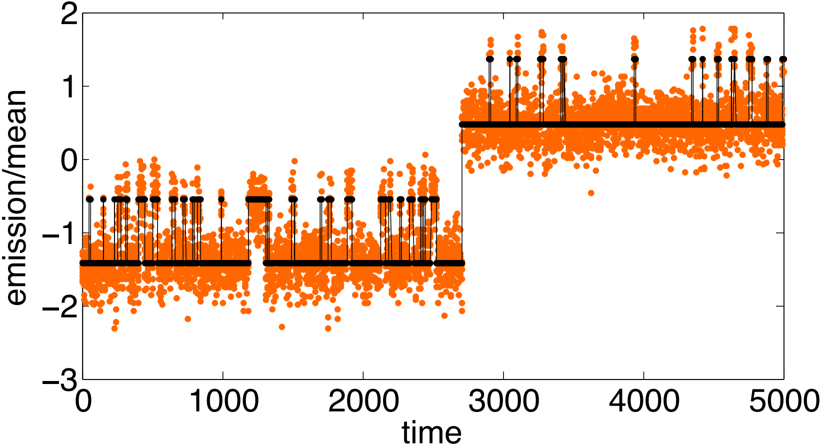
SIHSMM: Synthetic Data, State Duration Parameter Posterior



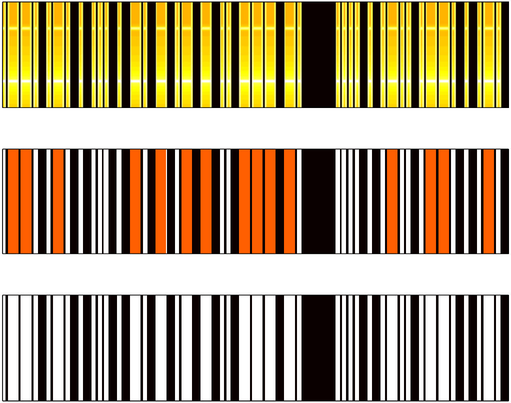
SIHSMM: Synthetic Data, State Mean Posterior



SIHSMM: Nanoscale Transistor Spontaneous Voltage Fluctuation

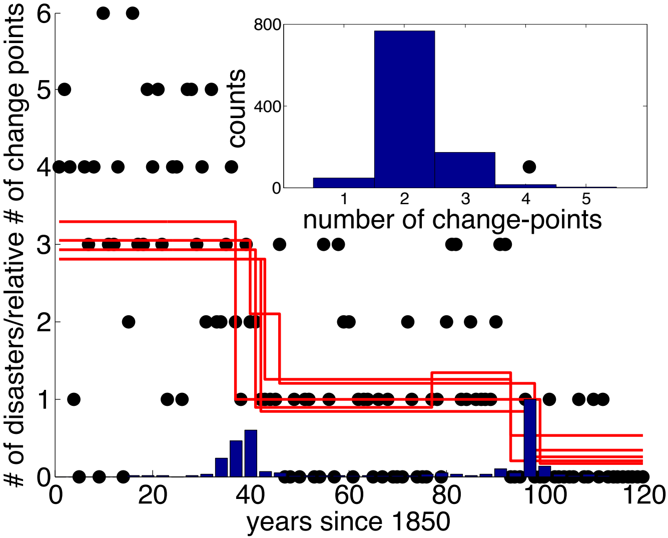


SIHSMM vs. IHMM: Modeling the Morse Code Cepstrum



Application from [15]

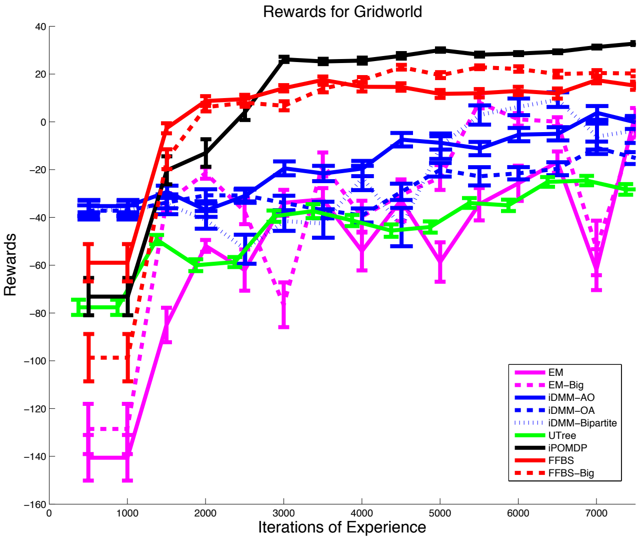
SIHSMM → Left-to-Right Infinite HSMM



Lessons and Questions

- Improving prediction must require latent variable models, right?
    - Do we bias towards the right ones?
    - Do existing priors probably penalize model complexity too harshly?
    - Existing latent variable models generally don't smooth like the SM.
        - How do we build models like this?
        - Would introducing such smoothing help prediction?
- Do we need more data/time?
    - What is the regime in which complex models will surpass simpler models for in-painting-like tasks?
    - Is there a convergence rate theory we're missing?
- Can we "smooth" downward? How?
- Early observation: award maximization and predictive loss minimization seem to result in opposite orderings

Comparative Predictive Performance for Reinforcement Learning



From [Doshi-Velez, Pfau, Wood and Roy, in sub.]

The End

Thank you.

References I

[1] N. Bartlett and F. Wood. Streaming deplump. In *Data Compression Conference 2011*, 2011.

[2] Nicholas Bartlett, David Pfau, and Frank Wood. Forgetting counts: Constant memory inference for a dependent hierarchical Pitman-Yor process. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 63–70, June 2010. URL `http://www.icml2010.org/papers/549.pdf`.

[3] Matthew J Beal, Z Ghahramani, and C E Rasmussen. The Infinite Hidden Markov Model. *Advances in Neural Information Processing Systems*, 1:577–584, 2002. ISSN 10495258. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.16.2929&amp;rep=rep1&amp;type=pdf`.

[4] M Dewar, C Wiggins, and F Wood. Inference in hidden Markov models with explicit state duration distributions. *In Submission*, 2011.

References II

[5] F. Doshi-Velez, D. Pfau, F. Wood, and N. Roy. Bayesian nonparametric methods for partially-observable reinforcement learning. In Sub., 2012.

[6] P. Dupont, F. Denis, and Y. Esposito. Links between probabilistic automata and hidden Markov models: probability distributions, learning models and induction algorithms. *Pattern recognition*, 38(9): 1349–1371, 2005.

[7] T.S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[8] E B Fox, E B Sudderth, M I Jordan, and A S Willsky. A Sticky HDP-HMM with Application to Speaker Diarization. *Annals of Applied Statistics*, 5(2A):1020–1056, 2011.

[9] J. Gasthaus and Y. W. Teh. Improvements to the sequence memoizer. In *Proceedings of Neural Information Processing Systems*, pages 685–693, 2011.

References III

[10] J. Gasthaus, F. Wood, and Y. W. Teh. Lossless compression based on the Sequence Memoizer. In *Data Compression Conference 2010*, pages 337–345, 2010.

[11] Ricard Gavald, Joelle Pineau, and Doina Precup. Pac-learning of markov models with hidden state. In *In Proceedings of ECML*, 2006.

[12] N. D. Goodman, V. K. Mansinghka, D. Roy, K. Bonawitz, and J. B. Tenenbaum. Church: a language for generative models. In *Uncertainty and Artificial Intelligence*, 2008. to appear.

[13] M. W. Ho, L. F. James, and J. W. Lau. Coagulation fragmentation laws induced by general coagulations of two-parameter Poisson-Dirichlet processes. http://arxiv.org/abs/math.PR/0601608, 2006.

[14] J. Huggins and F. Wood. Infinite structured explicit duration hidden Markov models. 2012.

References IV

[15] M Johnson and A S Willsky. The hierarchical Dirichlet process hidden semi-Markov model. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 252–259, 2010.

[16] M. Johnson, T.L. Griffiths, and S. Goldwater. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems*, volume 19, page 641, 2007.

[17] J F C Kingman. *Poisson Processes*. Oxford Studies in Probability. Oxford University Press, 1993.

[18] P. Liang, S. Petrov, M. Jordan, and D. Klein. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 688–697, 2007.

References V

[19] W Neiswanger and F Wood. Unsupervised detection and tracking of multiple objects with dependent Dirichlet process mixtures. In Sub., 2012.

[20] T.B. Paige, X. Zhang, J. Forde, and F. Wood. Perspective inference for eye-to-eye videoconferencing: Empirical evaluation tools and data. In Sub., 2012.

[21] David Pfau, Nicholas Bartlett, and Frank Wood. Probabilistic deterministic infinite automata. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1930–1938. MIT Press, 2010.

[22] David Pfau, N Bartlett, and F Wood. Probabilistic Deterministic Infinite Automata. In *Advances in Neural Information Processing Systems*, 2011.

References VI

[23] J. Pitman. Coalescents with multiple collisions. *Annals of Probability*, 27:1870–1902, 1999.

[24] V Rao and Y W Whye Teh. Spatial Normalized Gamma Processes. In *Advances in Neural Information Processing Systems*, pages 1554–1562, 2009.

[25] Simeon Realov and Kenneth L Shepard. Random Telegraph Noise in 45-nm CMOS : Analysis Using an On- Chip Test and Measurement System. *Analysis*, (212):624–627, 2010. ISSN 01631918. URL `http://dx.doi.org/10.1109/IEDM.2010.5703436`.

[26] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[27] C.E. Shannon. Prediction and entropy of printed english. *The Bell System Technical Journal*, pages 50–64, 1951.

References VII

[28] Y W Teh, M I Jordan, M J Beal, and D M Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476): 1566–1581, 2006.

[29] J Van Gael, Y Saatci, Y W Teh, and Z Ghahramani. Beam sampling for the infinite hidden Markov model. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1088–1095. ACM, 2008.

[30] F. Wood, C. Archambeau, J. Gasthaus, L. James, and Y. W. Teh. A stochastic memoizer for sequence data. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1129–1136, Montreal, Canada, 2009.

[31] F. Wood, J. Gasthaus, C. Archambeau, L. James, and Y. W. Teh. The sequence memoizer. *Communications of the ACM*, 2011.

Thoughts and Questions

- Bayesian nonparametric estimators fundamentally have the wrong storage asymptotics.
  - Should we abandon them?
  - Is the class of parametric estimators derived from approximate nonparametric estimators interesting?
  - Is there a provable advantage to or optimal strategy for forgetting in Bayesian nonparametric models?
- How do we distinguish between the expressivity of models of infinite complexity?
- How do we learn σ, or, more generally, how to we compute in a model with countable back-offs?
- What happens when the input vocabulary gets very big and the typical input vector is high-dimensional?

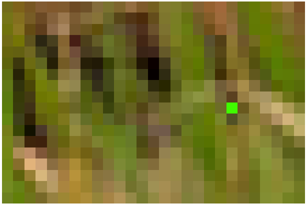
Musings

Models like the SM are tough to beat in in-painting-type applications because only local specificity matters.

- Text in-painting

  She put it ____ the bank.

- Image in-painting



Hypothesis : The Right Latent Variable World State Representation Should Matter

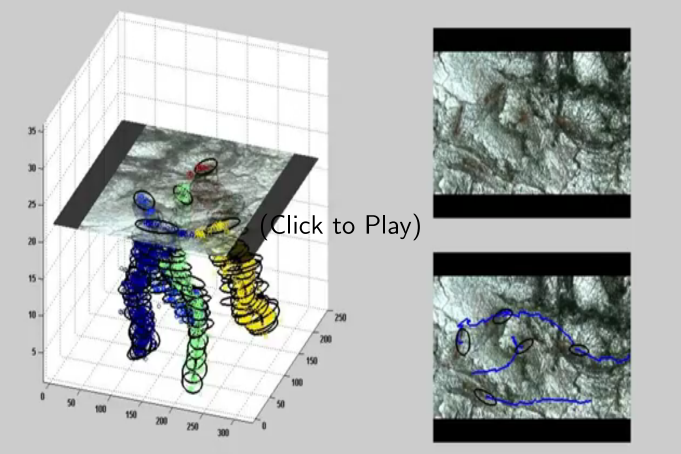
- Text in-painting
  - She just pulled a fish from the river.
  - She just received her allowance.

She put it ____ the bank.

- Image in-painting

For planning, arguably, only the latent state matters.

Ant Tracking



(Click to Play)

Figure : Multiple target tracking (ANTS) [Neiswanger and Wood, in sub.]

Object Detection and Tracking

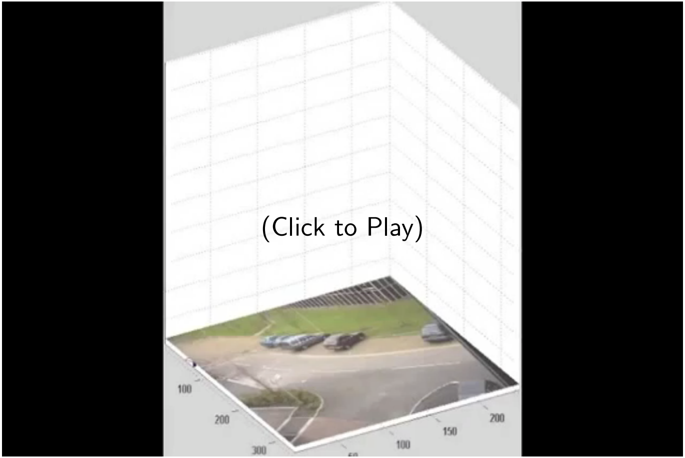

(Click to Play)
100
200
300
50
100
150
200

Figure : Multiple target tracking (PETS2000) [Neiswanger and Wood, in sub.]

Virtual Perspective Inference

(Click t

Figure : icareti [Brooks, Zhang, Forde, and Wood, in sub.]